

A

Date: July 21, 2000

JC857 U.S. PTO
09/620615
07/21/00

NEW APPLICATION TRANSMITTAL

Transmitted herewith for filing is the patent application of Inventor(s):

Tommy Kay Teague

For:

METHOD OF PROVIDING DUPLICATE ORIGINAL FILE COPIES OF A SEARCHED TOPIC FROM MULTIPLE FILE TYPES DERIVED FROM THE WEB

Enclosed are:

1. The Papers Required for Filing Date Under 37 CFR 1.53(b):

<u>34</u>	Pages of specification
<u>1</u>	Pages of Abstract
<u>6</u>	Pages of claims
<u>11</u>	Sheets of drawings
	_____ formal
	X informal

In addition to the above papers, there is also attached:

Pages of an amendment _____

CERTIFICATION UNDER 37 CFR 1.10

I hereby certify that this paper (along with any paper referred to as being attached or enclosed) is being deposited with the United States Postal Service on this date, July 21, 2000, in an envelope as "Express Mail Post Office to Addressee" Mailing Label Number EL070486690US addressed to: Box Patent Application, Assistant Commissioner for Patents, Washington, D.C. 20231.

Marion Comer De Yoe

(Type or print name of person mailing paper)

Marion Comer Di Goe
(Signature of person mailing paper)

NOTE: Each paper or fee referred to as enclosed herein has the number of the "Express Mail" mailing label placed thereon prior to mailing. 37 CFR 1.10(b).

_____ for processing an application with
a specification in a non-English language
(\$20.00; 37 CFR 1.53(d) and 1.17(k) \$ _____
_____ processing and retention fee
\$100.00; 37 CFR 1.21(l) \$ _____

NOTE: 37 CFR 1.21(l) establishes a fee for processing and retaining any application which is abandoned for failing to complete the application pursuant to 37 CFR 1.53(d) and this, as well as the changes to 37 CFR 1.53 and 1.78 indicate that in order to obtain the benefit of a prior U.S. application, either the basic filing fee or the processing and retention fee of \$1.21(l) within one year from notification under § 53(d) must be paid.

Total fees enclosed \$ 730.00

9. Method of Payment of Fees

_____ check in the amount of \$ _____
 X charge Account No. 19-3790 in the amount of \$730.00. A duplicate of this transmittal is attached.

NOTE: Fees should be itemized in such a manner that it is clear for which purpose the fees are paid. 37 CFR 1.22(b)

10. Authorization to Charge Additional Fees

NOTE: If no fees are to be paid on filing, the following items should **not** be completed.

WARNING: If these boxes are to be checked, then accurately count claims, especially multiple dependent claims, to avoid unexpected high charges.

 X The Commissioner is hereby authorized to charge the following additional fees which may be require to Account No. 19-3790 .

 X 37 CFR 1.16 (filing fees and presentation of extra claims)
_____ 37 CFR 1.17 (application processing fees)
_____ 37 CFR 1.18 (issue fee at or before Mailing of Notice of Allowance,
pursuant to 37 CFR 1.31(b).

NOTE: 37 CFR 1.28(b) requires "Notification of any change in loss of entitlement to small entity status must be filed in the application ...prior to pay...issue fee."

11. Instructions As To Overpayment

 X credit Account No. 19-3790
_____ refund

Alfred W. Kozak
Signature of Attorney

Alfred W. Kozak
(Type or print name of attorney)

Unisys Corporation
10850 Via Frontera, MS 1000
San Diego, CA 92127

Reg. No. 24,265

Tel No. (858) 451-4615

2. Declaration or oath

 X Enclosed
 X original
executed by (check all applicable boxes)
 X inventors(s)
 legal representative of inventor(s) 37 CFR 1.42 or 1.43
 joint inventor or person showing a proprietary interest on behalf
of inventor who refused to sign or cannot be reached.
 This is the petition required by 37 CFR 1.47 and the statement
required by 37 CFR 1.47 also attached. See item 7 below for
fee.

 Not enclosed

NOTE: Where the filing is a completion in the U.S. of an international application under 35 U.S.C. 371(c)(4), the declaration must be filed.

 Application is made by a person authorized under 37 CFR 1.41(c) on behalf of all
of the above-named inventors(s). The declaration or oath, along with the surcharge
required by 37 CFR 1.16(e) can be filed subsequently.

NOTE: It is important that all the correct inventors(s) are named for filing under CFR 1.41(c) and 1.53(b).

 Showing that the filing is authorized. (Not required unless called into question.
37 CFR 1.141(d).

3. Language

 X English
 non-English

NOTE: An application including a signed oath or declaration may be filed in a language other than English. A verified English translation of the non-English language application and the processing fee of \$20.00 required by 37 CFR 1.17(K) is required to be filed with the application or within such time as may be set by the Office. 37 CFR 1.52(d).

NOTE: A non-English oath or declaration in the form provided or approved by the PTO need not be translated. 37CFR1.60(b).

 A verified English translation of the specification is attached.

4. Assignment

 X An assignment of the invention to UNISYS CORPORATION

5. Certified Copy

 A certified copy of application(s) from which priority is claimed.

NOTE: Must be referred to in oath or declaration. 37CFR1.55 and 1.63.

6. Fee Calculation

CLAIMS AS FILED

Number Filed	Number Extra	Rate	Basic Fee \$690.00
Total Claims -11- -20=	-0- X	\$ 18.00	-0-
Independent Claims -3- -3=	-0- X	\$ 78.00	-0-
Multiple dependent claim(s) if any -0-	-0- X	\$260.00	-0-

_____ Amendment canceling extra claims enclosed
 _____ Amendment deleting multiple dependencies enclosed
 _____ Fee for extra claims not being paid at this time

NOTE: If the fees for extra claims are not paid on filing, they must be paid or the claims canceled by amendment prior to the expiration of time period set for response by the Patent and Trademark Office in any notice of fee deficiency.

37 CFR 1.16(d)

Filing Fee Calculation \$ 690.00

7. Small Entity Statement

_____ verified statement that this is a filing by a small entity under 37 CFR 1.9 and 1.27 is attached.

Filing Fee Calculation (50% of above) \$ _____

8. Fee Payment Being Made At This Time

_____ Not Enclosed

_____ No filing fee is to be paid at this time. This and the surcharge required by 37 CFR 1.6(e) can be paid subsequently.

NOTE: Where the filing is completion in the U.S. of an international application, the fee must be paid.

 X Enclosed

 X basic filing fee \$ 690.00

 X recording assignment \$ 40.00

(40.00 37CFR 1.21(h)(j))

_____ petition fee for filing by other than
 all the inventors or person on behalf
 of the inventor where inventor refused to sign
 or cannot be reached. (\$130.00 37 CFR 1.47 and 1.17(h)) \$ _____

**TITLE: METHOD OF PROVIDING DUPLICATE ORIGINAL FILE
 COPIES OF A SEARCHED TOPIC FROM MULTIPLE FILE
 TYPES DERIVED FROM THE WEB**

FIELD OF THE INVENTION:

The present disclosure involves methods for developing full text searches for searching multiple file types which are downloaded from the Web.

CROSS-REFERENCES TO RELATED APPLICATIONS:

This application is related to a co-pending application, USSN _____ entitled "Method For Searching Multiple File Types on a CD-ROM", which is
5 incorporated herein by reference.

BACKGROUND OF THE INVENTION:

In present day commercial situations, many digital development software and computer companies work to deliver documentation to their customers in a number
10 of different formats. These formats may show up in a number of different varieties, that is to say the document format may be on paper, for example, or Adobe Acrobat Portable Document Format (PDF) files, or Windows Help files, or Hypertext Markup Language (HTML) and also
15 HTML help files.

The documentation provided to receivers, such as customers, is distributed and made available on, for example, paper documents, on CD ROMs, and on Web Servers.

Of course, it is desirable for a recipient or
20 user to make a full text search of the received documents. However, users cannot perform full-text searches on paper documents, except through long, laborious reading and surveys of the documents. There is, however, software designated as "search engines" that
25 exist in digital technology in order to search files that are distributed to users who download from the Web.

However, these search engines are limited in a number of ways in providing search capability when the document or received Web files involve multiple file
30 types. Most of the existing search engines are designed only to search files of one particular format.

In this type of situation, then it would be necessary to convert all files in the Web documents or Web-received files into a common format. This common format would be the format which was compatible with the particular search engine available.

However, when files are converted into a format different from that in which they were originally created, much of the functionality for searching the original file is lost, and this includes navigating through the file and finding certain special graphics or other content in the file.

There are other types of search engines which are capable in a certain limited way of including search operations for multiple file types in the Web received file documentation. However, these search engines are unable to open all the file types at locations where the search terms appear and then be capable of moving from one such location to the next location within the document.

Thus, these other types of search engines require that the user first search with one particularly favorite engine and then refine the search using another search engine designed for the file type.

One example of a standard (not a full-text) search is what one can do in a product program such as Word. The operator tells Word to find a text string. Then Word starts reading the text in the document by reading each word one at a time beginning at a specified location and comparing the text against the string that was entered. Now, when Word finds a "hit" (match), then Word highlights the text and stops searching. If the operator chooses "Find Next" option, then the Word program repeats the process and continues the search beginning just past the current hit. However, this is considered pretty much of a brute force and very slow process of operation.

A "full text" search, however, works to search a collection of files at one time. It accomplishes this by using an auxiliary collection of files that was created ahead of time and then distributed with the files that are to be searched. If, for example, the operator wished to search 450 files for the word "server," the software would then read the auxiliary files which will already know all occurrences and locations of the word "server." Here the software would present the operator with a "hit list" of all files that contained the word that is built from the information in the auxiliary files. If the operator elects to open up any of these files, the software will then open the file, move to the first location in the file (which it already knows from the auxiliary file), and then highlight the word. It may be noted that none of the files are directly searched or scanned. By using such a file, the operator or user can utilize advanced features such as wild cards ("install *") and Boolean operators ("installation and not printers").

There are a number of ways to create these auxiliary files. Such a process may take several hours for most of releases to be made on CD-ROM. The success of a "search engine" can be measured by how efficiently the desired files are generated and accessed.

The present invention provides for the use of an existing search engine that is designed to support the searching of one particular file format (PDF, or Adobe® Acrobat® files). This can then be extended to allow the searching of virtually any other type of file format such as HTML, HTML Help, or Windows Help. The method and system accomplishes this by creating a PDF file "duplicate" consisting of the text from the file that the operator wants to search in order to allow the search engine to find the text in the duplicate that was created. Here then there is provided a link from each

awk\appl\041503L.doc

SUMMARY OF THE INVENTION:

The described method involves the handling of multiple files downloaded from the web which files may exist in quite different word formats which are not readily searchable for desired topics or word matches.

The present method and system involves a technique that converts the downloaded file types into a Portable Document Format which uses an Adobe Acrobat program to search Portable Document Format (PDF) files that contain the text extracted from files residing in other formats such as Windows Help, Hypertext Markup Language (HTML) Help, and HTML.

On each page of the PDF file there are hyperlinks that the user can select to open the original file at the corresponding location.

The method enables the user to search the collection of PDF files, including both files that were created as PDF files as well as the PDF files created from the text extracted from the files of other formats. The method uses the search engine from Verity that is distributed by Adobe® in order to search the Adobe® Acrobat® portable document format files (PDF) which were downloaded from the Web. If the search targets include files of formats other than PDF, then the user is presented with pages within the PDF copy of the file in which the target text appears.

The user can navigate within the PDF copy using the "next hit" and "previous hit" program options. The text is visible to the user and is sufficient to help the user determine whether it is necessary or helpful to access the original file.

Each page of the PDF file carries a "button" that, when selected, opens the document in the original format at the location corresponding to the location displayed in the PDF copy. Both the PDF copy and the

original file are accessible at the same time so it is possible to identify the location of the hits within the file and to find additional hits in the complete collection of files.

5 The indicated method includes software which is used to extract the text from Windows Help, HTML, and HTML Help files, and then create from that text the new files that can be converted by the standard Adobe software into PDF files with corresponding explanatory
10 messages and buttons on every page in order to support the linking into the corresponding locations within the original files.

 This method then provides the ability to link from the hits displayed in Adobe Acrobat into the
15 corresponding locations within the original files.

BRIEF DESCRIPTION OF THE DRAWINGS:

Fig. 1A is a block diagram illustrating the environmental modules utilized in downloading files from the web for later conversion and search operations;

5 Figure 1B is a generalized schematic drawing showing how files in various formats are converted by a utility program into Portable Document Format (PDF) files;

10 Figure 2 is a schematic flowchart showing the method in searching non-portable document format files;

 Figure 3 is a representation of a window which indicates messages to the operator for finding other matches;

15 Figure 4 is a drawing showing the basic steps involved in converting files from various different formats into PDF files and then linking them to desired portions of the original file;

20 Fig. 5 is a flow chart illustrating the conversion of a Windows Help File into Rich Text Format (RTF);

 Fig. 6 is a flow chart illustrating the conversion of HTML files to Rich Text Format (RTF);

 Fig. 7 is a flow chart showing the conversion of an HTML Help file to Rich Text Format (RTF);

25 Fig. 8 is a flow chart showing the conversion of a Rich Text Format file to Portable Document Format (PDF) files;

30 Fig. 9 is a flow chart illustrating a search which can be instituted on the PDF files after multiple file types have been converted to PDF;

 Fig. 10 is a set of selected topic files side-by-side indicating one topic file in PDF copy format and the same topic file in original copy format.

GLOSSARY LIST

ACTIVE X CONTROL: This is Windows software. It often has a visual element, either at design time or run time. ActiveX controls also have the ability to communicate
5 some other program types, such as Microsoft Internet Explorer.

ACROBAT: This is document exchange software from Adobe Systems Incorporated of Mountain View, California that runs on DOS, Windows, Unix, and Macintosh computers. It
10 allows documents created on one platform to be displayed and printed exactly the same on another platform. Documents are converted into the Acrobat PDF (Portable Document Format) which contains all the information about the appearance of the document.

ADOBE ACROBAT DISTILLER: This is a software program that is part of the Adobe Acrobat suite which converts a PostScript file into a PDF file.
15

ADOBE ACROBAT PROGRAM: This is a software suite which facilitates the creation and access of PDF files. Adobe
20 Systems Incorporated, 345 Park Avenue, San Jose, CA 95110-2704.

ADOBE SOFTWARE CONVERTER: This is a software program that extracts text from a Windows Help, HTML, or HTML Help and creates an RTF file.

BUTTON: This is one of several kinds of interface items that can be displayed on a dialog by a Windows program. A command button is chosen by the user to begin, interrupt, or end a process. When chosen, a command button appears pushed in, and is sometimes called a "push button."
25

CD-ROM (Compact Disk-Read Only Memory): This is a compact disk format used to hold text, graphics, and even
30

high fidelity stereo sound. It is similar to an audio compact disk but uses a different track format for data. The audio CD player cannot play CD-ROMs, but CD-ROM players can usually play audio CDs. CD-ROMs hold in
5 excess of 600 megabytes of data which is equivalent to about 250,000 pages of text or approximately 20,000 medium-resolution images.

CHM FILE: This is a Compiled Help file. This type of file is supported by Microsoft to replace Windows Help
10 files.

CLIPBOARD: A temporary memory storage location supported by Microsoft Windows which allows a user to transfer text, graphics, code, etc., from one application to another.

15 ENGINE: This is the portion of the program that determines how the program manages and manipulates data. An engine differs from a user interface, with which the user communicates with the program, and it differs from other parts of a program, such as installation routines
20 and device drivers, which enable the program to use a computer system and its components. The term "engine" is rarely used on its own and is more often mentioned in relationship to a particular program. For example, a database engine is the portion of a database management
25 program that contains the tools for manipulating a database. A search engine would be that part of a program used to search and find a particular digital word or coded index.

FILE FORMAT: The structure of a file that defines the
30 way it is stored and laid out on the screen or in print. The format can be fairly simple and common, as are the files stored as plain ASCII text, or it can be quite complex and include various types of control instructions

and codes used by programs and by printers or other devices. Examples of formats include RTF (Rich Text Format); DCA (Document Content Architecture); PICT, DIF (data interchange format), DXF, TIFF (tagged image file format), and EPSF (Encapsulated PostScript Format).

FORMAT: This involves a structure or layout of an item. Screened formats are fields on the screen; report formats are columns, headers and footers on a page. Record formats are the fields within a record. File formats are the structure of data and program files, word processing documents and graphics files (display lists and bitmaps) with all their proprietary headers and codes.

FORMAT PROGRAM: This is software that initializes a disk. There are two formatting levels. The low level initializes the disk surface by creating the physical tracks and storing sector identifications in them. Low level format programs lay out the sectors as required by the particular type of drive technology used (IDE, SCSI, etc.). The high-level format creates the indexes used by the operating system (Mac, DOS, etc.) to keep track of the data stored in the sectors.

FULL-TEXT SEARCH: Full-Text search is a mechanism for searching for text in a collection of documents using various criteria. Adobe makes this available for files released on CD-ROM and Verity for files released on Web sites. It is necessary in both these cases to create auxiliary files to support full-text search. The user may search all documents or any subset of the documents using wildcards--for example, searching for "install*" will find all occurrences of install, installing, installation, installed, etc. The user may also use Boolean arguments--for example, searching for "installation and printers" will find all documents in which both the words "installation" and "printers" occur.

Contrast full-text search with a simple find, in which the software scans all text in the document from the beginning looking for the indicated literal text.

HTM: This is a file name extension -- for example, CONTENTS.HTM or INDEX.HTM. This extension is usually used to identify files read by an Internet browser, such as Internet Explorer or Netscape.

HTM EXTENSION: This is a Windows/DOS file name extension equal to HTM. For example, CONTENTS.HTM or INDEX.HTM. This extension is usually used to identify files ready by an Internet browser, such as Internet Explorer or Netscape.

HTML (Hypertext Markup Language): This is a standard for defining hypertext links between documents. It is a subset of SGML (Standardized General Markup Language).

HTML HELP: Microsoft HTML Help is the standard help format for Windows 98 and Windows 2000. It is much more capable than standard HTML, since it provides sophisticated features such as Dynamic HTML and ActiveX controls.

HYPERLINK: A hyperlink is a part of a page, whether the page is displayed from a CD-ROM or from a Web site, that the user can click with the mouse to perform some function, such as open a document, play a video, or display an external file.

HYPERTEXT: This is linking related information. For example, by selecting a word in a sentence, information about that word is retrieved if it exists, or the next occurrence of the word is found. This is also a metaphor for presenting information in which text, images, sounds, and actions become linked together in a complex, non-sequential web of associations that permit the user to

browse through related topics regardless of the presented order of the topics. These links are often established both by the author of a hypertext document and by the user, depending on the intent of the hypertext document.

5 For example, traveling among the links to the word "iron" in an article might lead the user to the periodic table of the elements or else a map of the migration of metallurgy in iron age Europe. The term "hypertext" was coined to described documents (as presented by a
10 computer) that expressed the non-linear structure of ideas as opposed to the linear format of books, films, and speech.

INNERTEXT METHOD: This is a software mechanism to invoke the procedure called InnerText within the Microsoft
15 ActiveX control that supports Internet Explorer. Extracts unformatted text from within the body of an HTML file.

NEXT HIT OPTION: This is an option provided by a search engine to facilitate navigation from one "hit," or found
20 item, to the next. Ordinarily, the user performs a search and the search engine presents the user with a "hit" list. This is a list of documents in which the items for which the user is searching can be found. When the user opens a document from the list, the first "hit"
25 in the document is displayed. The user then moves to successive hits by selecting the next hit option.

ORIGINAL FILE: The concept of original file applies to the process described by this disclosure. In this case, it would be the Windows Help, HTML, or HTML Help file
30 that is created to be released with the application. A utility reads the original file and creates a companion PDF file that consists of the unformatted text from the original file.

ORIGINAL PDF: This is a PDF file that was originally created to be delivered as a PDF file. It is usually a complete book, and it includes all graphics, special fonts, etc.

- 5 PDF COPY: This is a PDF file that was created from another type of file, such as Windows Help, HTML, or HTML Help. It contains only the text from the other file.

- 10 PDF FILES CREATED FROM TEXT EXTRACTED FROM OTHER FILE TYPES: The disclosure includes utilities that read the unformatted text from other types of files. The text is used to generate a PDF companion file of the original file that has links from each page into the corresponding location within the original file.

- 15 POSTSCRIPT DRIVER: This is Windows software which facilitates printing from a Windows application to a PostScript printer.

- 20 POSTSCRIPT FILE: This is a Windows file created by redirecting the commands generated by a PostScript driver to a file instead of to a printer. It can be copied to a PostScript printer or used by Adobe Acrobat Distiller to produce PDF files.

- 25 PREVIOUS HIT OPTION: This is an option provided by a search engine to facilitate navigation from one "hit," or found item, to the next. Ordinarily, the user performs a search and the search engine presents the user with a "hit" list. This is a list of documents in which the items for which the user is searching can be found. When the user opens a document from this list, the first "hit" in the document is displayed. The user then moves to
- 30 successive hits by selecting the next hit option. Once the user has selected the next hit option, it is possible to return to the previous successive hit by selecting the previous hit option.

RTF: This is Rich Text Format, an adaptation of DCA (Document Content Architecture). This allows a user to transfer formatted text documents between applications, even those running on different platforms.

- 5 RTF FILE IN WORD: This is the process of opening an RTF file in Word. Word converts the RTF file into a Word document.

- 10 RTF PAGES: These are pages displayed in Word when it has an RTF file open. This allows the developer to see the separate pages.

- 15 SEARCH: This is the action of seeking the location of a file, or to search a file or data structure for specific data. A search is carried out by comparison or calculation to determine whether a match to some specified pattern exists or whether some other criteria have been met.

SEARCH ALGORITHM: This is an algorithm designed to locate a particular element, called a target in a list.

- 20 SEARCH TARGET: The search target is the text which defines what is being searched for. This could be a literal string of text which is to be found, such as "installation instructions," or a string containing wildcards, such as "install*", or a string containing Boolean instructions, such as "installation and
25 printers."

SEARCH TERM: See "Search Target."

- 30 SENDKEYS: This is a function supported by Visual Basic and some other programs running under Windows that permits one software application to send keystrokes to another to simulate user input.

UNFORMATTED TEXT: This term refers to text that does not contain formatting information attributes, such as font name, point size, bold, italics, underline, etc., or does not possess the structure associated with tables, columns, indented paragraphs, etc.

VERITY SEARCH ENGINE: This is a software suite developed by Verity, and used on the Unisys Support Web site, that facilitates full-text search of files on a Web site. It includes both the software that the site administrator has to execute to create files necessary to support full-text search as well as the software that the user accesses to perform the searches. Verity Inc., 894 Ross Drive, Sunnyvale, CA 94089.

WEB BROWSER: A client application that enables a user to view HTML documents on the World Wide Web, another network, or the user's computer; follow the hyperlinks among them; and transfer files. Text-based Web browsers, such as Lynx, can serve users with shell accounts but show only the text elements of an HTML document: most Web browsers, however, require a connection that can handle IP packets but will also display graphics that are in the document, play audio and video files, and execute small programs, such as Java applets or ActiveX controls, that can be embedded in HTML documents. Some Web browsers require helper applications or plug-ins to accomplish one or more of these tasks. In addition, most current Web browser permit users to send and receive e-mail and to read and respond to newsgroups.

WINDOWS: This is an operating system introduced by Microsoft Corporation in 1983. Windows is a multi-tasking graphical user interface environment that runs on MS-DOS based computers. Windows provides a standard interface based on drop-down menus, windowed regions on the screen, and a pointing device such as a mouse. The

programs used must be specially designed to take advantage of these features. A graphics-based operating system from Microsoft that provides a desktop environment similar to the Macintosh in which applications are displayed in re-sizeable moveable windows on a screen. Starting with Windows 95, the Windows system is a self-contained 32-bit operation system that requires a minimum Intel 386. In order to use all the features of Windows, applications must be written for this system.

- 10 WINDOWS HELP: Windows-based help systems are automated Windows utilities that provide procedural and system information to software users in lieu of paper-based documentation. Windows-based help supports context-sensitive help, which lets the user access topics in a help file that are relevant to the user's location in the application.

DESCRIPTION OF PREFERRED EMBODIMENT:

Fig. 1A is a generalized drawing which illustrates the environmental modules which constitute the operating modules which permit the conversion of
 5 downloaded multiple-type files from the Web into Portable Document Format (PDF) files for observation on a observable window by the operator.

Now referring to Fig. 1A, a personal computer
 10 is seen having a memory 12 and operating system 14 and is also connected to a disk storage unit 16.

The personal computer 10 (user workstation) is provided with an Adobe Acrobat program 22.

The World Wide Web 5 is seen connected to the personal computer 10 and may download digital data in
 15 various different formats.

A Verity Search Engine 9 connected to the terminal server 8 can initiate a search on the Web 5 and bring about a download of multiple files to the user workstation 10. However, some of these files may be in
 20 one particular format, while others may be in different formats, thus instigating a problem when a browser or search engine is used in order to find a particular subject matter or topic on any one of the particular files.

Fig. 1B is an overall generalized drawing showing the basic steps in the creation of text copies from various types of downloaded files for conversion into Portable Document Format, or PDF files. For example, as seen in Fig. 1A, the Windows Help file (W1)
 30 is converted by a utility program (U2) into a Portable Document Format copy designated (WC).

Again, in Fig. 1A, a hypertext mark-up language file (HTML) designated as (M1) is passed through a utility program (U2M) after which there is provided at

step (MC) a Portable Document Format copy of this particular file.

Further, in Fig. 1A, there is seen an HTML Help file (HH1) which is passed through a utility program (U2HH) in order to provide a Portable Document Format copy designated (HHC).

The original PDF file is designated as Opdf. This is the PDF file that was originally created to be delivered as a PDF file. It is usually a complete book, and includes all the graphic, special fonts, charts and other special arrangements, etc.

Now referring to Fig. 2, there is seen a generalized view for the searching of non-Portable Document Format files. Here, it is desired that a search be made on a particular topic or target such as "I/O" for example, in order to finally provide and display the data of the original file on that particular topic. Thus, as seen in Fig. 2, at step (NP1), there is instituted a search of all of the Portable Document Format (PDF) files.

Then, at step (NP2), the program will navigate to a particular page in the Portable Document File (PDF).

At step (NP3), the operator can click a button which appears on that particular page that is displayed, and then at step (NP4), the operator can open the original file to the selected topic, for example, such that the original target topic, such as "I/O" will now be displayed and seen in its original file form.

Fig. 3 is a schematic drawing of a window which can be observed by the operator which can be found on the Acrobat Reader tool bar in regarding to finding other matches.

Seen on this window is a set of icons, one of which can be pressed for "search" and another icon which can be pressed for search results. Then, there is another icon which shows a way to find the previous match

and highlight the previous match, in addition to an icon used to find the next match and highlight the next match.

The search results icon will provide a display of a list of documents that contain matches, while the
 5 search icon is used to change the search topics.

Fig. 4 is a slightly more detailed drawing of sets of flow charts showing the basic steps involved in converting files from various different formats into PDF files and then with subsequent linking of these files to
 10 desired portions of the original file.

A sequence of original files are shown which are to be the object of a search. The Windows Help files are designated W1 and the HTML files are designated M1, while the HTML Help files are designated HH1, and the
 15 Help file is designated H1.

The next step involved respectively, for each of these files is the extraction of text. This is shown respectively, as block W2, M2, HH2, and H2, which represents in each case the factor of extracting the text
 20 of a particular topic or target subject matter.

The next level of steps shown respectively, as W3, M3, HH3, and H3, all involve the step of conversion with use of the Adobe Acrobat software converter.

Then, the next respective sequence of steps
 25 involves steps W4, M4, HH4, and H4 which involve the development of the Portable Document Format, or PDF files.

Then in Fig. 4, there is seen step W5 which involves two separate functions, one of which is the set
 30 of buffers to hold the PDF files, together with an explanation message regarding the files in the buffer. An example of an explanation message and a link created by this program are shown in the left panel of Figure 10.

Then at step W6, a link occurs from the
 35 explanation message and buffers of step W5 in order to provide for step W7 which locates and displays the

appropriate section of the original file on the topic matter that was desired.

As will be seen in the next succeeding set of drawings, it should be understood that there are certain
 5 intermediate steps involved, whereby the original files are first converted to Rich Text Format (RTF), after which the subsequent RTF files can then later be converted to Portable Document Format (PDF).

Now, there is seen in Fig. 5 which shows the
 10 various steps in flow chart form, for converting the Windows Help file to Rich Text Format. Starting at step W1, the program will acquire the name of the Input Windows Help file and the name of the Output Rich Text Format file.

15 At step W2, the program will open the Windows Help file.

At step W3, the program will initiate a utility to report the count of topics and topic IDs. A Windows Help file is composed of a collection of individual
 20 topics. Every topic has a number, from 1 through the total number of topics. Each topic can have a Topic ID: for example, "Using Boolean Expressions in Acrobat Searches". This step generates a list which is used by subsequent steps in the process to read every topic in
 25 the Windows Help file that has a topic ID.

At step W4, the program will then go to the list to read the number of the next topic that has a Topic ID. For example, this next topic might be the subject of "Channel Adapters".

30 At step W5, a decision block is presented to query whether or not additional topics are present. If there are no additional topics, then the program will end at step W5E. On the other hand, if a topic is present (YES), then step W6 occurs where the program will use
 35 SENDKEYS to the Windows Help file to open the topic up and copy the text from that topic into the Clipboard.

Then at step W7, the program will copy the text from the Clipboard and format the Rich Text Format pages, after which there is a return to step W4 in order to get the text from the next topic.

5 Fig. 6 is a flow chart illustrating the steps involved for converting the HTML files to Rich Text Format (RTF). At step 1, the program will acquire the name of the directory containing the HTML files and also the name of the Output Rich Text Format (RTF) file. Note
10 that an HTML "document" can consist of a number of files with the HTM extension.

Then at step M2, the program will get the next file in the directory with the HTM extension. This is a Windows/DOS file name extension, which is equivalent to
15 HTM, as for example, CONTENTS.HTM or INDEX.HTM. This extension is usually used to identify files read by an Internet browser, such as Internet Explorer or by Netscape.

At step M3, a decision block is presented which
20 presents the query as to whether or not another file with the HTM extension is present. If the answer is (NO), then the program will end at step M3E. If the answer is (YES) at step M3, then step M4 occurs to open the particular file with the ActiveX control which will use
25 the InnerText method to read the text. InnerText is a software mechanism within the Microsoft ActiveX control that supports Internet Explorer and will extract unformatted text from within the body of a HTML file.

Then, at step M5, the program will format the
30 Text into Rich Text Format pages (RTF).

After step M5, the program loops back to step M2 to get the next file in the directory with the HTM extension.

35 Fig. 7 is a flow chart illustrating the conversion of an HTML Help file into a Rich Text Format (RTF) file. An HTML Help file is also called a CHM file

or a compiled Help file. This is a type of file supported by Microsoft and used to replace Windows Help files. A CHM file is constructed from a collection of HTML files.

Here at step HH1, the program will acquire
 5 names of the CHM file directory, which contains the HTML files from which the CHM file is constructed and the Output RTF file to be created by the program.

At step HH2, the program will get the next file
 10 in a directory with the HTM extension. The extension is used to identify files read by an Internet browser.

At step HH3, a query block is presented to
 query whether an additional file with an HTM extension is present. If the answer is (NO), then the program ends here at step HHE. If the answer is (YES), that is to
 15 say, a file is present, then at step HH4, the program will open the file with the ActiveX control and use the InnerText method to read the text. This copies unformatted text from within the body of a HTML file. Graphics, font information, such as point size, bold, italic, etc., and structure, such as tables, columns,
 20 etc., are not copied.

Then at step HH5, the extracted text is operated on to format the text into Rich Text Format (RTF) pages.

25 After this, the program loops from HH5 back to HH2 in order to operate on the next file in the directory.

As was previously discussed, the Rich Text Format files are a kind of intermediate file which
 30 eventually must be converted to a portable document format, or PDF file. Fig. 8 is a flow chart showing the steps involved for converting the Rich Text Format file to the Portable Document File.

At step CRP1, the program will open the Rich
 35 Text Format file in Word so that the Word program of

Microsoft will convert the Rich Text Format file into a Word document.

At step CRP2, the program will use the Word program to print to file, using a PostScript driver. The
5 PostScript driver is a portion of Windows software which facilitates printing from a Windows application to a PostScript printer.

At step CRP3, there is developed a PostScript file which is a Windows file created by redirecting the
10 commands generated by a PostScript driver to a file, instead of to a printer. The file can be copied subsequently to a PostScript printer or just used by the Adobe Acrobat Distiller to produce Portable Document Format files.

At step CRP4, the program will open the PostScript file in the Adobe Acrobat Distiller.

At step CRP5, the program will use the Adobe Acrobat Distiller to produce the Portable Document Format files.

With the development of the PDF file as shown in Fig. 8, the Portable Document File can now relate to Fig. 4 which shows the level of Portable Document Format files seen at steps W4, M4, HH4, and H4.

Then, as was illustrated in Fig. 4 through
25 steps W5, W6 and W7, the files are placed in buffers with an explanation message and then linked to the appropriate sections of the original file for display of the topic material in its original format with all its graphics, lists, drawings, and any unusual factors that appeared in
30 the original file.

This can further be expounded by the flow chart seen in Fig. 9, where now that the Portable Document Format (PDF) copies have now been isolated, then a search can be initiated using the Adobe Acrobat programs.

Now referring to Fig. 9 at step S1, the program will initiate a search of a particular topic through the Adobe Acrobat program.

Then at step S2, there is presented a list of
 5 the Portable Document Format (PDF) documents, showing the list of hits to the user.

At step S3, the user selects a Portable Document Format document and opens it to the first hit.

At step S4, a decision box is initiated to
 10 query of whether the file is originally a Portable Document File. If the answer is (YES), then the program sequence is to step S7 to query whether the search should end.

At step S4, if the answer is (NO), that is to
 15 say, the file is not originally a Portable Document Format file, then at step S5 the user will click the "Open Document" button on the top of the display page.

At step S6, the original document is now opened to the particular topic containing the text in the
 20 Portable Document Format file.

At step S7, a decision box presents the question of whether this is the end of the search. If the answer is (YES), the search ends at step S7E. If it is not the end of the search (NO), then step 8 occurs
 25 where the user clicks the "next hit" button on the tool bar of the Portable Document Format file.

Then, step S8 loops back to step S4 in order to continue through S5, S6 and S7 until the search has ended at S7E.

Now referring to Fig. 10, there is illustrated
 30 a page of unformatted text which is shown on the left side of the page, and its corresponding original file which is indicated on the right-hand side of the page.

As an example, the subject matter was that of
 35 "Establishing a named pipe to a COMs Application". Here, it will be noticed that the unformatted text does not

contain all the information, such as graphics, etc., but that the original file shown on the right-hand side shows the original text together with the graphics and detailed material which may not appear in the unformatted text.

5 Thus, it can now be understood that a series of document information such as articles, books or manuals can be downloaded from the Web and exist in different types of formats. This normally would make it unwieldy or impossible to search through the entire list of
10 downloaded documents in order to get information on a particular topic that was desired since any one particular search browser is specific to the handling of any one particular format, but not available or useful in handling the many different format types involved, or
15 multiple types of formats.

 Thus, the present system, by using the intermediate step of providing the Rich Text Format which can then be converted to the Portable Document Format, and then the Portable Document Format is utilized as
20 being compatible with and accessible to search purposes by use of the Adobe Acrobat program, the multiple numbers of different files, documents, articles or pages downloaded from the Web via the Verity Search Engine can now be searched for a given topic and then displayed in
25 Portable Document Format (PDF).

 Then subsequently, the Portable Document Format (PDF) can then be linked back to the original text of the original pages holding the desired topic information desired by the user and these can be displayed in their
30 original format with full graphics, colors, lists, tables and any other types of display which would not be available in the PDF format.

 While a particular implementation of the above-described invention has been shown in a particular
35 effective implementation, there may be other implementations of the invention which are derivable from

awk\appl\041503L.doc

WHAT IS CLAIMED IS:

1. A system for searching the web for targeted Websites and downloading the targeted document files to a user-terminal for a topic search comprising:

5 (a) terminal server means for searching the Internet on targeted Websites;

(b) user-terminal means to download said targeted Websites as document files to a user-terminal means;

10 (c) means for converting said document files into a common format;

(d) search means for searching said downloaded common format document files for a designated topic;

15 (e) opening means to view those document files indicated as a hit;

(f) selection mans to enable viewing of each said common-format downloaded pages holding the designated topic which was indicated as a hit.

2. The system of claim 1 which includes:

(g) means to link said downloaded page to a copy of the downloaded originally formatted page holding the selected search topic.

3. The system of claim 1 wherein said terminal server means includes:

5 (a1) Verity search engine means for targeting a selected Website on the Internet.

4. The system of claim 1 wherein said search means (d) for searching said downloaded document files includes:

(d1) means to convert said downloaded document files to a common format;

5 (d2) means to search said downloaded document files in their common format for a selected topic.

5. The system of claim 4 which includes:

(d3) means for selecting pages having hits for viewing by a user.

6. The system of claim 5 which includes:

(d4) means for utilizing a "next hit" or "previous hit" button to view new listed hit pages on the selected topic.

7. The system of claim 4 wherein said means (c) for converting downloaded document files to a common format includes:

5 (c1) auxiliary utility program means for enabling conversions of multiple file formats to a common Portable Document Format (PDF).

8. The system of claim 7 wherein said search means (d) for searching said downloaded document files includes:

5 (d5) Adobe Acrobat program means for searching said common format PDF files for a selected topic and generating a list of hits.

9. The system of claim 4 which includes:

(f) linking means for viewing the original page of a selected topic which correlates to the selected topic page in said common format.

10. A system for accessing multiple formatted document files from the World Wide Web for subsequent topic searching and viewing the hits comprising:

- 5 (a) terminal server means for searching targeted Websites and downloading selected hits of target documents;
- (b) conversion means to convert said downloaded hit document files to a common format;
- 10 (c means for searching said common format downloaded hit as document files for a selected topic;
- (d) means for utilizing those common formatted document hit pages for viewing by a user;
- 15 (e) means for linking said common format hit pages to the original hit pages for viewing by a user.

11. A method for searching and viewing selected-topic pages of variously formatted document files downloaded from the Web, comprising the steps of:

5 (a) searching the World Wide Web for selected Websites;

(b) downloading said selected Websites which represent hits to access multiple document files which occur in many different formats;

10 (c) converting each of said multiple document files to a common format;

(d) searching said common format document files with a compatible search engine to provide a directory of document files which have hits;

15 (e) displaying for view, each page of said commonly formatted document files having hits;

(f) linking each commonly formatted page of document hits to the corresponding original document for viewing the pages of said original document.

ABSTRACT OF THE DISCLOSURE:

**TITLE: METHOD OF PROVIDING DUPLICATE ORIGINAL FILE
 COPIES OF A SEARCHED TOPIC FROM MULTIPLE FILE
 TYPES DERIVED FROM THE WEB**

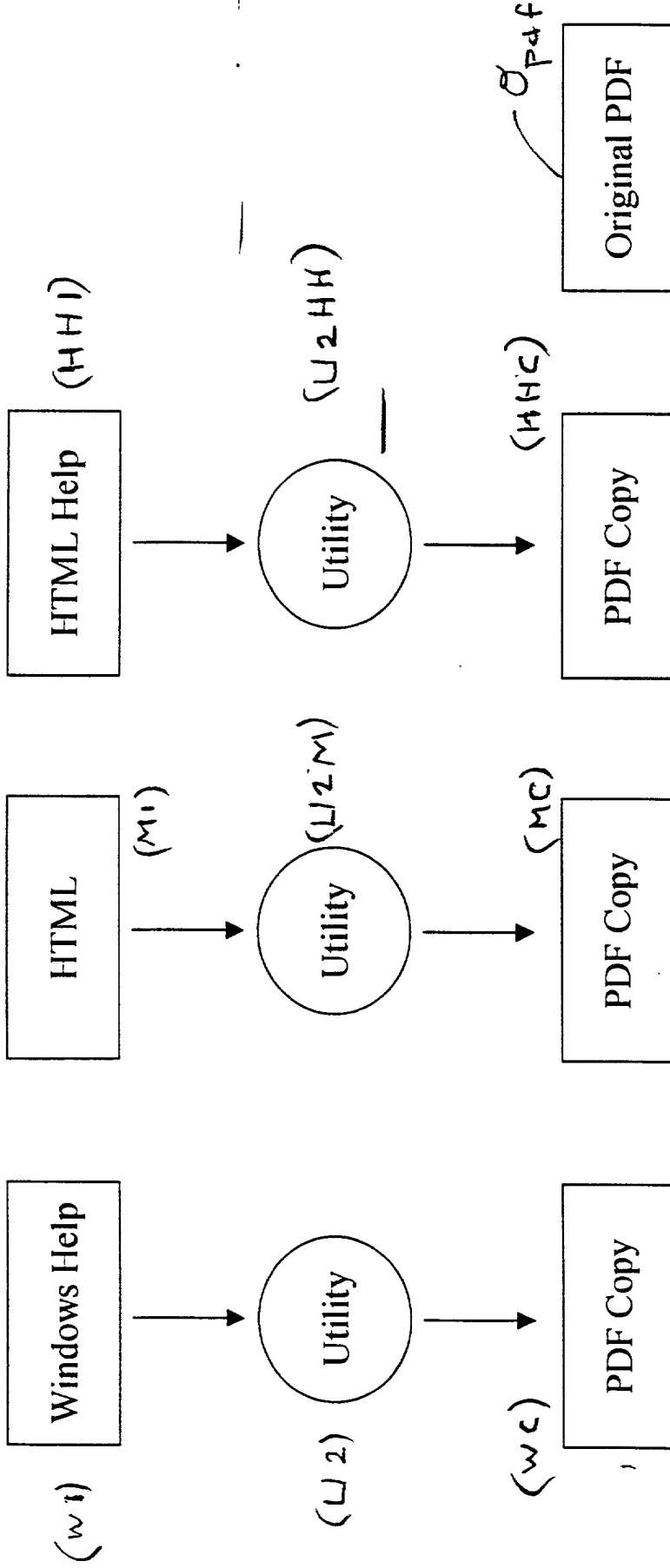
Many document files in different formats can be downloaded from websites which can be selected for their specific content using search items as a Verity Search Engine and Web Server. After downloading into a user-
5 workstation, a topic search would not be ordinarily feasible to search files of different formats. The present system and method enables topic searching by converting the different file formats into a common format such as PDF which then can easily be topic-
10 searched by a browser such as an Adobe Acrobat program.

The diagram illustrates a system architecture with the following components and connections:

- SERVER (8):** Represented by a laptop icon. It is connected to the **WEB (5)** and the **VERIFY SEARCH ENGINE (9)**.
- VERIFY SEARCH ENGINE (9):** A rectangular box connected to the **SERVER (8)** via a bidirectional arrow.
- WEB (5):** A rectangular box with a tilde symbol (~) below the number 5. It is connected to both the **SERVER (8)** and the **USER WORKSTATION (10)**.
- USER WORKSTATION (10):** Represented by a laptop icon. It is connected to the **WEB (5)**, **ADOBE ACROBAT (22)**, **MEMORY (12)**, **OP. SYST. (14)**, and **DISK (16)**.
- ADOBE ACROBAT (22):** A rectangular box connected to the **USER WORKSTATION (10)** via a bidirectional arrow.
- MEMORY (12) and OP. SYST. (14):** Two stacked rectangular boxes connected to the **USER WORKSTATION (10)** via bidirectional arrows.
- DISK (16):** A cylindrical storage device connected to the **USER WORKSTATION (10)** via a bidirectional arrow.

FIG. 1A

DKT 503-L



Creation of Text Copies as PDF

FIG 1F

6/12/00

DKT 241-503-L

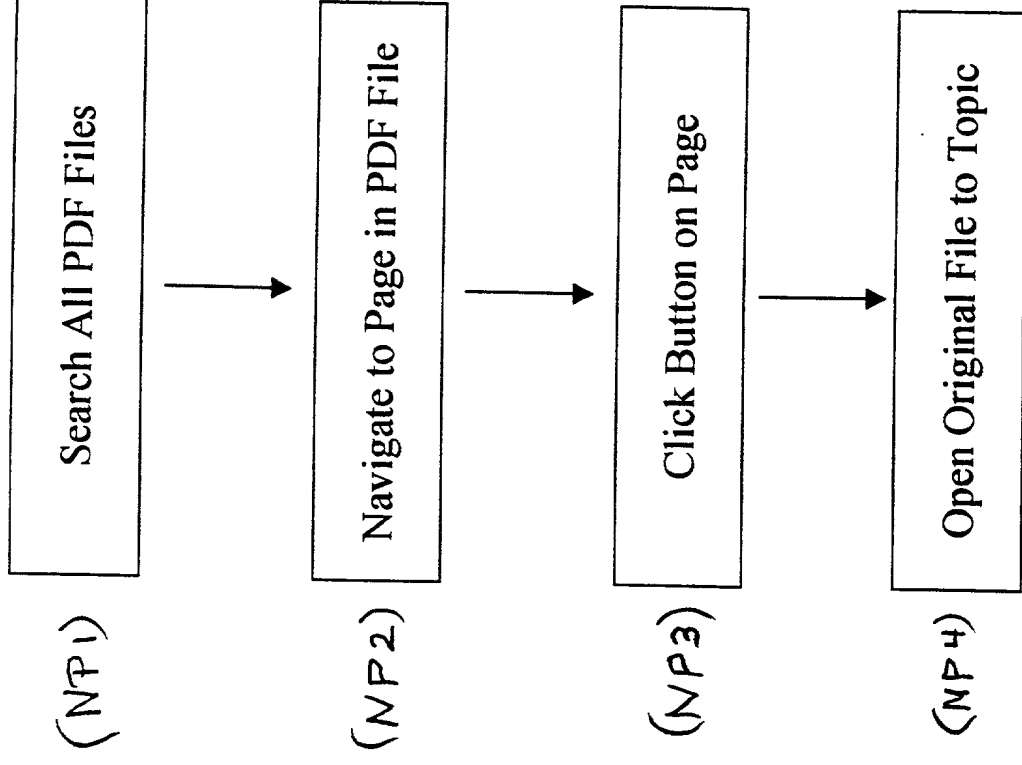


FIG. 2

Searching Non-PDF Files

The diagram illustrates the search interface. At the top, there is a 'Search' box and a 'Search Results' box. Below the 'Search' box is a row of icons: a magnifying glass, a document with a magnifying glass, a document with a magnifying glass and a checkmark, a document with a magnifying glass and a cross, and a document with a magnifying glass and a magnifying glass. Below the 'Search Results' box is a row of icons: a magnifying glass, a document with a magnifying glass, a document with a magnifying glass and a checkmark, a document with a magnifying glass and a cross, and a document with a magnifying glass and a magnifying glass. Below the row of icons are two boxes: 'Previous Highlight' and 'Next Highlight'.

Click:

Next Highlight

Previous Highlight

Search Results

Search

OK

FIG. 3

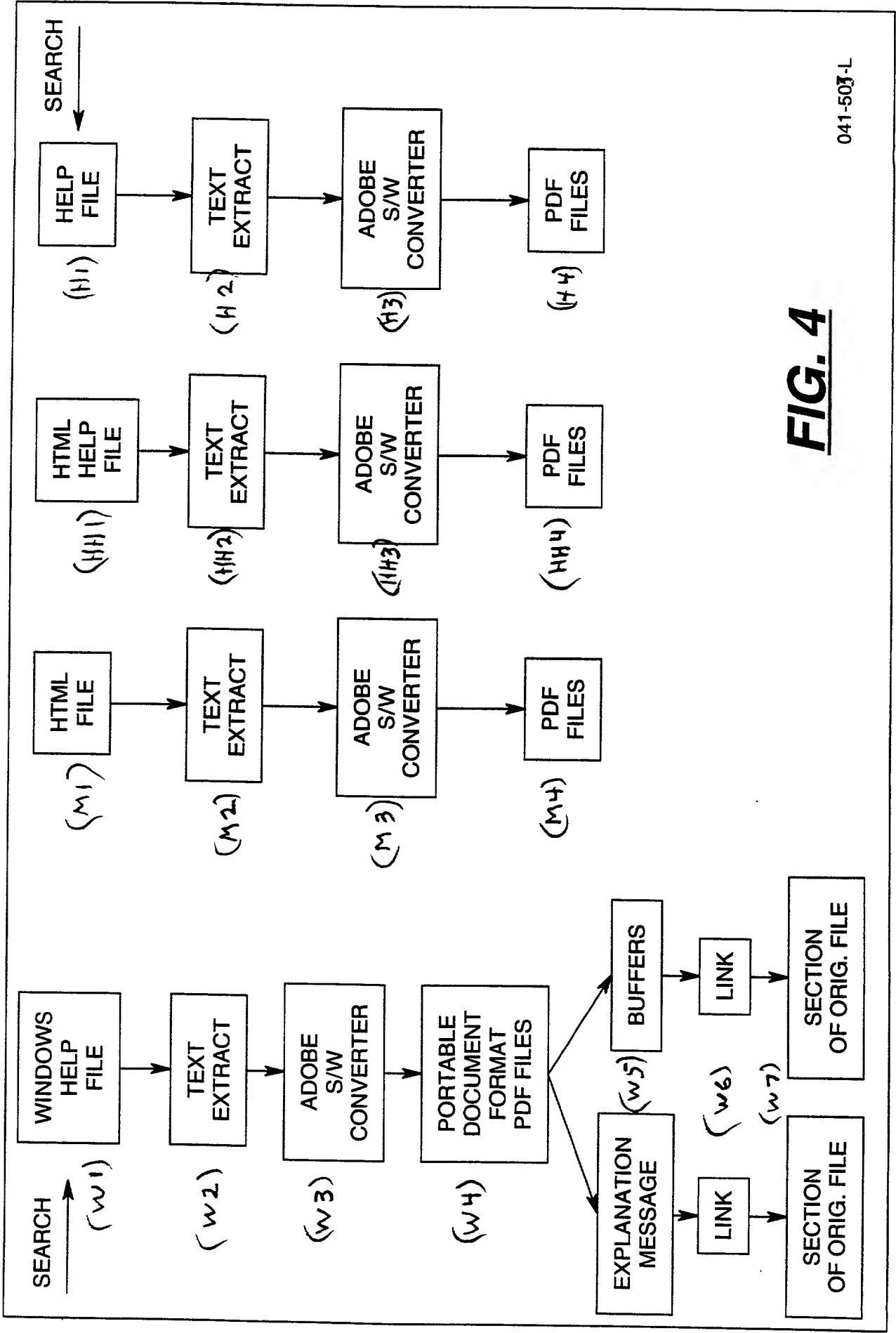
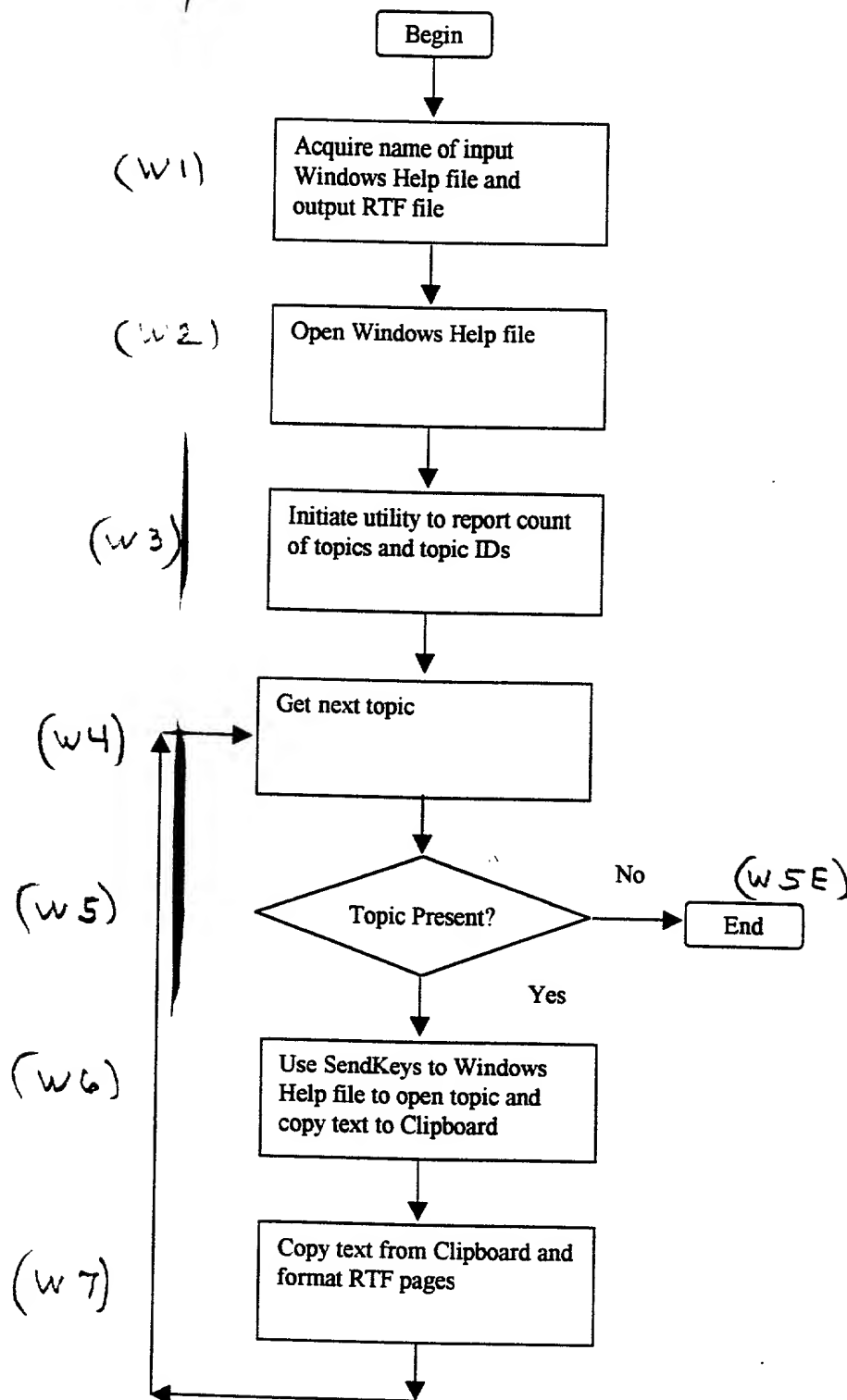
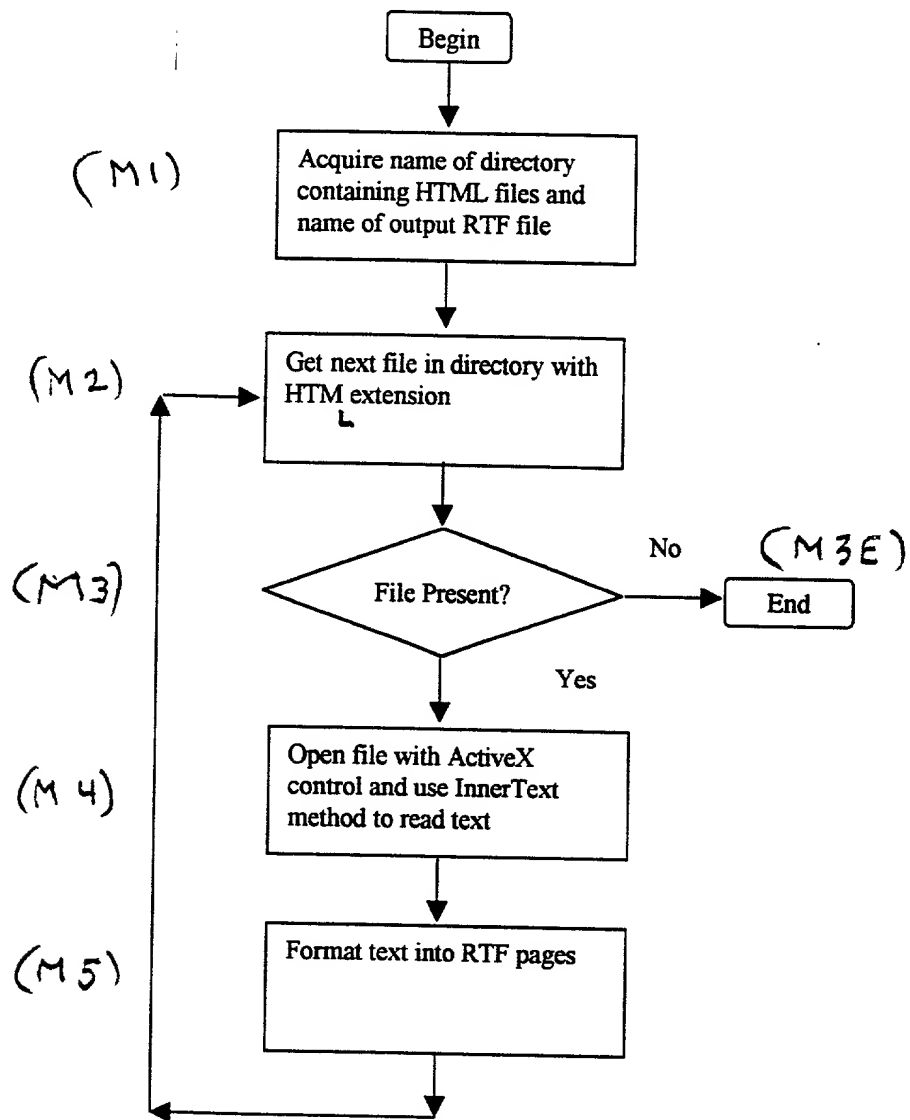


FIG. 4



Converting Windows Help File to RTF

FIG. 5



Converting HTML Files to RTF

FIG. 6

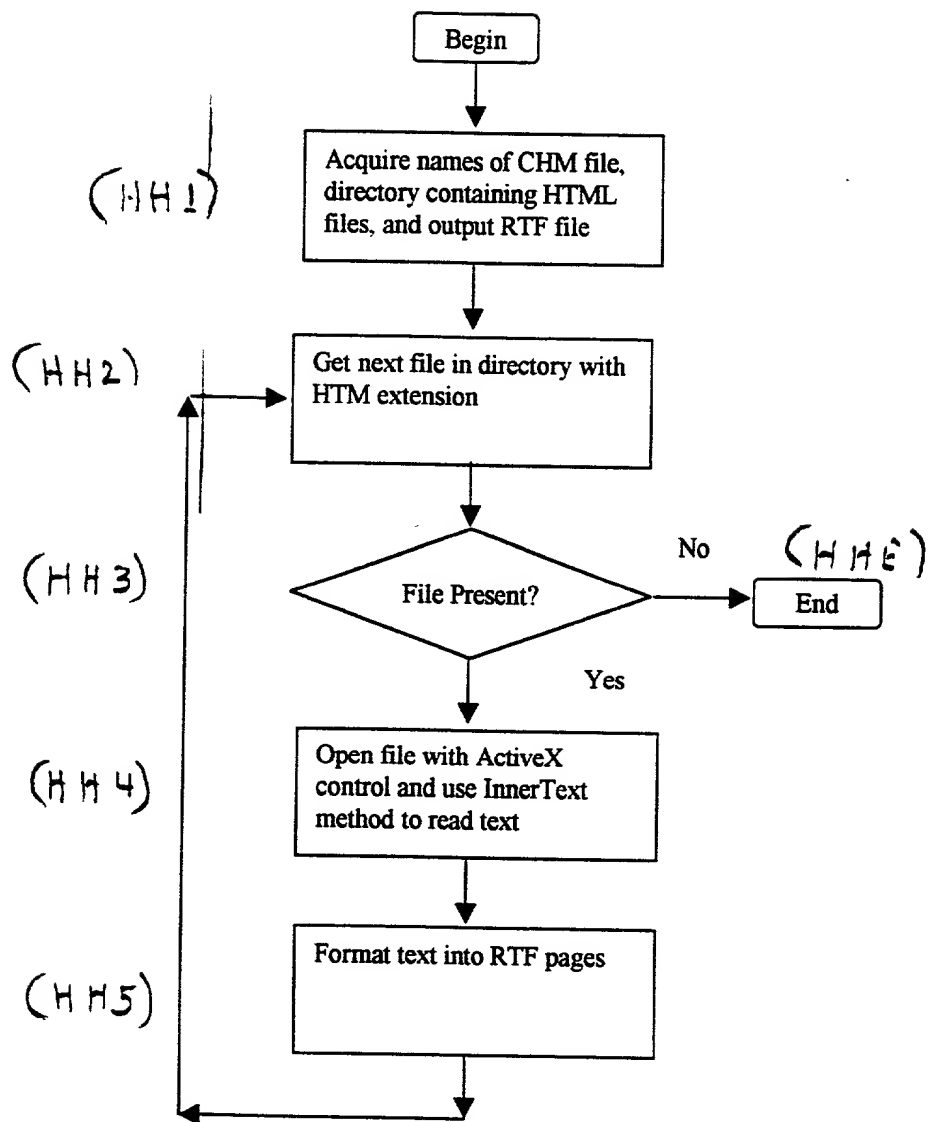


FIG. 7

Converting HTML Help (CHM) Files to RTF

This page contains
unformatted text.

Open
Topic

Establishing a Named Pipe to a COMS Application

Establishing a Named Pipe to a COMS Application

Note: This functionality is applicable to ClearPath servers only.
To establish a named pipe to a COMS application, a client program opens a named pipe of the following form:

Notice the first three nodes of the named pipes filename are fixed: \\<server>\PIPE\COMS. If the 4th node and beyond is a <Pipes PCM template>, then the resulting dialog's service attribute (that is, the next CCF service in the connection) is specified by the template's service attribute. If this attribute is undefined, then the first node of the <Pipes PCM template> name is used as the next CCF service in the connection. If a template exists having an asterisk as its last character, this character is treated as a wildcard. This causes an association with any named pipes filename that matches the characters preceding the asterisk. When the wildcard templates conflict, the template with the most characters takes precedence. A template of only an asterisk becomes the default for named pipes files that do not match a specified filename.

Example:

A template of PAYROLL* would be applied to the named pipe
\\SRV1\PIPE\COMS\PAYROLLPAYWIND and
\\SRV1\PIPE\COMS\PAYROLLPAYWINDSTAABC

If the 4th and subsequent nodes do not match to a <Pipes PCM template>, the 4th node itself is considered to be a <CCF service>. In this case, the resulting dialog's service attribute (that is, the next CCF service in the connection) is this <CCF service>. This connection then uses the Pipes PCM default template (identified by a *) for its connection attributes. No attributes are currently defined for this default template, but that doesn't prevent it from being modified.

Other connection attributes are gleaned directly from NX/Services. These attributes include: UserCode, ComputerName, Domain, PCUser, and IPAddress

Establishing a Named Pipe to a COMS Application

Note: This functionality is applicable to ClearPath servers only.

To establish a named pipe to a COMS application, a client program opens a named pipe of the following form:

\\<server>\PIPE\COMS\ <CCF service> | <Pipes PCM template> | \<COMS

Notice the first three nodes of the named pipes filename are fixed: \\<server>\PIPE\COMS. If the 4th node and beyond is a <Pipes PCM template>, then the resulting dialog's service attribute (that is, the next CCF service in the connection) is specified by the template's service attribute. If this attribute is undefined, then the first node of the <Pipes PCM template> name is used as the next CCF service in the connection. If a template exists having an asterisk as its last character, this character is treated as a wildcard. This causes an association with any named pipes filename that matches the characters preceding the asterisk. When the wildcard templates conflict, the template with the most characters takes precedence. A template of only an asterisk becomes the default for named pipes files that do not match a specified filename.

Example:

UNFORMATTED TEXT

ORIGINAL FILE

FIG. 10

Attorney's Docket No. 041-503-L

COMBINED DECLARATION AND POWER OF ATTORNEY

As a below named inventor, I hereby declare that:

TYPE OF DECLARATION

This declaration is of the following type:

- ☒ original
- ☐ design
- ☐ supplemental
- ☐ divisional
- ☐ continuation
- ☐ continuation-in-part (CIP)

INVENTORSHIP IDENTIFICATION

My residence, post office address and citizenship are as stated below next to my name, I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which are a patent is sought on the invention entitled:

TITLE OF INVENTION

METHOD OF PROVIDING DUPLICATE ORIGINAL FILE COPIES OF A SEARCHED TOPIC FROM MULTIPLE FILE TYPES DERIVED FROM THE WEB

SPECIFICATION IDENTIFICATION

the specification of which: (complete (a), (b) or (c))

- (a) ☒ is attached hereto.
- (b) ☐ was filed on _____ as ☐ Serial No.
or ☐ Express Mail No., as Serial No. not yet known

ACKNOWLEDGEMENT OF REVIEW OF PAPERS AND DUTY OF CANDOR

I hereby state that I reviewed and understand the contents of the above identified specification, including the claims, as amended by an amendment referred to above.

I acknowledge the duty to disclose information

- which is material to patentability as defined in 37, Code of Federal Regulations, § 1.56
- and which is material to the examination of this application, namely, information where there is a substantial likelihood that a reasonable examiner would consider it important in deciding whether to allow the application to issue as a patent, and

— In compliance with this duty there is attached an information disclosure statement in accordance with 37 CFR 1.98.

POWER OF ATTORNEY

I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith.

ALFRED W. KOZAK, REG. NO. 24,265
MARK T. STARR, REG. NO. 28,762

SEND CORRESPONDENCE TO

ALFRED W. KOZAK
UNISYS CORPORATION
10850 VIA FRONTERA, MS 1000
SAN DIEGO, CALIFORNIA 92127

DIRECT TELEPHONE CALLS TO:
(Name and telephone number)

ALFRED W. KOZAK
(858) 451-4615

DECLARATION

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

SIGNATURE(S)

Full name of sole or first inventor Tommy Kay Teague

Tommy
(GIVEN NAME)

Kay
(MIDDLE INITIAL OR NAME)

Teague
FAMILY (OR LAST NAME)

Inventor's signature Tommy Kay Teague

Date July 20, 2000 Country of Citizenship USA

Residence: 22942 Luciana, Mission Viejo, California 92691

Post Office Address same as above